

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2003-101900

(P2003-101900A)

(43) 公開日 平成15年4月4日(2003.4.4)

(51) Int.Cl. ⁷	識別記号	F I	キーワード* (参考)
H 0 4 N 5/45		H 0 4 N 5/45	5 C 0 2 3
H 0 4 B 1/16		H 0 4 B 1/16	C 5 C 0 2 5
			G 5 C 0 2 6
			Z 5 C 0 6 3
H 0 4 N 5/265		H 0 4 N 5/265	5 K 0 6 1

審査請求 未請求 請求項の数22 O L (全 20 頁) 最終頁に続く

(21) 出願番号 特願2001-287055(P2001-287055)

(22) 出願日 平成13年9月20日(2001.9.20)

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72) 発明者 山本 高司

東京都大田区下丸子3丁目30番2号キヤノン株式会社内

(72) 発明者 荒谷 俊太郎

東京都大田区下丸子3丁目30番2号キヤノン株式会社内

(74) 代理人 100090538

弁理士 西山 恵三 (外1名)

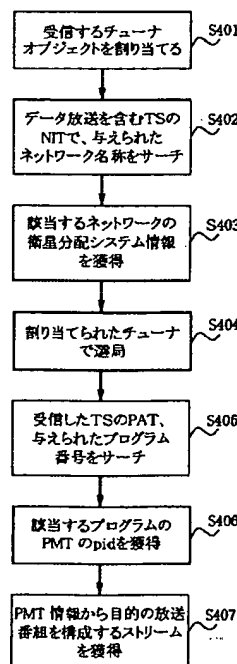
最終頁に続く

(54) 【発明の名称】 受信装置

(57) 【要約】

【課題】 簡単な操作にて複数の番組を同時に視聴可能とする。

【解決手段】 受信装置は、複数のチューナによりテレビジョン放送波を受信し、複数の映像信号及び制御信号を出力すると共に、前記複数の映像信号に係る映像を同一画面上に表示すべく前記複数の映像信号を合成処理して表示部に出力するにあたり、受信された制御信号に従い、前記複数のチューナにより受信すべきチャンネルと前記映像処理手段による表示レイアウトとを自動的に設定する構成とした。



【特許請求の範囲】

【請求項1】 複数のチャンネルを有するテレビジョン放送波を受信し、映像信号及び制御信号を出力する複数の受信手段と、

前記複数の受信手段により受信された複数の映像信号に係る映像を同一画面上に表示すべく前記複数の映像信号を合成処理して表示部に出力する映像処理手段と、前記受信手段より出力された制御信号に従って前記複数の受信手段と前記映像処理手段とを制御し、前記複数の受信手段により受信すべきチャンネルと前記映像処理手段による表示レイアウトとを自動的に設定する制御手段とを備える受信装置。

【請求項2】 前記制御信号は所定のチャンネルにて送信されることを特徴とする請求項1記載の受信装置。

【請求項3】 前記受信手段は更に放送番組の内容に関する電子番組データを出力し、前記電子番組データに基づいて前記所定のチャンネルを含む番組表を生成して前記表示部に表示する番組表生成手段を備えたことを特徴とする請求項2記載の受信装置。

【請求項4】 前記番組表生成手段は前記番組表内の前記制御信号の放送時間に応じた位置に前記制御信号に従い受信すべき番組の内容に関する情報を表示することを特徴とする請求項3記載の受信装置。

【請求項5】 前記制御信号は前記所定のチャンネル以外のチャンネルにて放送される放送番組の放送時間に応じて受信すべきチャンネル及び表示レイアウトが変化することを特徴とする請求項2記載の受信装置。

【請求項6】 前記制御手段は前記番組表において前記制御信号に係る放送番組が選択されたことに応じて、前記制御信号に従い前記受信手段と前記映像処理手段とを制御することを特徴とする請求項3から請求項5のいずれかに記載の受信装置。

【請求項7】 前記所定のチャンネルはBSデジタル放送におけるデータ放送チャンネルであり、前記制御信号は前記データ放送データとして送信されることを特徴とする請求項2記載の受信装置。

【請求項8】 前記制御信号に従い前記複数の映像信号に多重すべき文字情報あるいはグラフィック情報を生成して前記映像処理部に出力するグラフィック生成手段を備え、前記制御手段は更に、前記制御信号に従い前記映像処理手段を制御して前記文字情報あるいはグラフィック情報の表示位置を決定することを特徴とする請求項1記載の受信装置。

【請求項9】 前記制御手段は操作手段による表示画面上に表示された前記文字情報あるいは前記グラフィック情報に関連する操作に従い、前記複数の映像信号に係る前記表示レイアウトを変更することを特徴とする請求項8記載の受信装置。

【請求項10】 前記制御信号は前記文字情報あるいは前記グラフィック情報に関連する操作に従う前記表示レ

イアウトの変更動作を記述したデータを含むことを特徴とする請求項9記載の受信装置。

【請求項11】 前記表示レイアウトは前記複数の映像信号に係る映像の表示位置、解像度及びサイズのうちの少なくとも一つを含むことを特徴とする請求項1記載の受信装置。

【請求項12】 前記複数の受信手段は更に、前記映像信号に係る音声信号を出力し、前記制御手段は更に、前記制御信号に従い前記複数の受信手段より出力される音声信号の音声モニタ手段に対する出力動作を制御することを特徴とする請求項1記載の受信装置。

【請求項13】 データ放送チャンネルを含む複数のチャンネルを有するBSデジタル放送波を複数のチューナにより受信する装置であって、

前記複数のチューナにより受信された複数のチャンネルの映像信号に係る映像を同一画面上に表示すべく前記複数の映像信号を合成処理してモニタ部に出力する映像処理手段と、

前記チューナにより受信したデータ放送チャンネルより所定の制御テンプレートデータを検出し、この制御テンプレートデータに従って前記複数のチューナと前記映像処理手段を制御し、前記複数のチューナにより受信すべきチャンネルと前記映像処理手段による表示レイアウトとを自動的に設定する制御手段とを備えるテレビジョン受信機。

【請求項14】 複数のチャンネルを有するテレビジョン放送波を受信し、映像信号を出力する複数の受信手段と、

前記複数の受信手段により受信された複数の映像信号に係る映像を同一画面上に表示すべく前記複数の映像信号を合成処理して表示部に出力する映像処理手段と、装置外部より制御プログラムを入力する入力手段と、前記制御プログラムにより指定された複数のチャンネルを同時に受信すると共に、前記制御プログラムにて指定された表示レイアウトにて前記同時に受信されたチャンネルの映像信号に係る映像を表示するよう前記入力手段により入力された制御プログラムに従って前記複数の受信手段と前記映像処理手段とを制御する制御手段とを備える受信装置。

【請求項15】 前記制御手段は前記制御プログラムと時間経過とに従い前記受信チャンネルと表示レイアウトとを変更することを特徴とする請求項14記載の受信装置。

【請求項16】 前記入力手段はネットワークを介して前記制御プログラムを入力することを特徴とする請求項14記載の受信装置。

【請求項17】 前記入力手段は記憶メディアから前記制御プログラムを読み出すメディアドライブを含むことを特徴とする請求項14記載の受信装置。

【請求項18】 前記記憶メディアはディスク状記憶媒

体あるいはメモ리카ードを含むことを特徴とする請求項17記載の受信装置。

【請求項19】 前記複数の受信手段は更に、前記映像信号に係る音声信号を出力し、前記制御手段は更に、前記制御信号に従い前記複数の受信手段より出力される音声信号の音声モニタ手段に対する出力動作を制御することを特徴とする請求項14記載の受信装置。

【請求項20】 複数の受信手段により複数のチャンネルを有するテレビジョン放送波を受信し、映像信号及び制御信号を出力すると共に、前記複数の映像信号に係る映像を同一画面上に表示すべく前記複数の映像信号を合成処理して表示部に出力する方法であって、前記受信された制御信号に従い、前記複数の受信手段により受信すべきチャンネルと前記映像処理手段による表示レイアウトとを自動的に設定することを特徴とする受信方法。

【請求項21】 複数の受信手段により複数のチャンネルを有するテレビジョン放送波を受信し、映像信号を出力すると共に、前記複数の映像信号に係る映像を同一画面上に表示すべく前記複数の映像信号を合成処理して表示部に出力する方法であって、装置外部より制御プログラムを入力し、前記制御プログラムにより指定された複数のチャンネルを同時に受信すると共に、前記制御プログラムにて指定された表示レイアウトにて前記同時に受信されたチャンネルの映像信号に係る映像を表示するよう前記入力された制御プログラムに従って前記複数の受信手段と前記映像処理手段とを制御することを特徴とする受信方法。

【請求項22】 請求項20及び請求項21に記載の方法をコンピュータにより実現するためのプログラムを記憶した記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は受信装置に関し、特に、複数のチャンネルの映像を同時に受信して表示する装置に関する。

【0002】

【従来の技術】 従来、テレビ受信機において二台のチューナを搭載するなどして二つの放送番組を受信して同時に画面に表示する機能が知られている。このような機能により、ある放送番組を視聴しながら同じ時間帯に重なった別の放送番組も視聴することができる。

【0003】 一方、今日、ケーブルテレビジョン（CATV）、CS放送に加えてBSデジタル放送の開始により、従来とは比較にならない程多くのチャンネルにより放送が行われている。

【0004】 また、BSデジタル放送においては複数の映像により一つの放送番組を構成するマルチビュー放送が行われる。これを自由に視聴できるようテレビ受信機側も複数のチューナやデコーダを備え、より多くの画面

を同時に表示する機能を持つことが考えられる。こうした場合、テレビ受信機の表示画面のレイアウトも、従来のようにテレビ受信機に予め用意されているレイアウトだけではなく、より自由なレイアウトが望まれるようになる。

【0005】

【発明が解決しようとする課題】 このように多くのチャンネルにより非常に多数の番組が放送されるようになると、テレビ視聴の目的に適う番組がどの時間帯でどのチャンネルで放送されるかを正確に把握することが難しくなる。ユーザは視聴の目的に適う番組を探すため、従来から提供される雑誌、新聞に掲載される番組予定表を精査する事に加え、BSデジタル放送においては放送波に多重されている電子プログラムガイドを利用して番組予定も調べなければならない。さらにこれら多数の番組の中から同時に視聴すべき番組の組み合わせを探し出して視聴計画を立案するには、多大な労力と時間を要する。娯楽としてのテレビ視聴にそぐわない状況である。

【0006】 また、テレビ受信機の機能が高度化してより多くの画面を同時に表示できるようになり、表示レイアウトの柔軟性を備えてくると、このようなテレビ受信機の機能の多さに比例して複雑な操作を要するようになる。

【0007】 更に、視聴する番組の選択操作まで考慮すると快適な操作は困難な状況である。

【0008】 本発明はこのような問題点を解決し、簡単な操作にて複数の番組を同時に視聴可能とすることを目的とする。

【0009】 本発明の他の目的は、多数のチャンネルの中から視聴すべき番組を選択する操作や、複数の番組を同時に視聴する際の表示レイアウトの設定操作などの負担を軽減する処にある。

【0010】

【課題を解決するための手段】 前記課題を解決するため、本発明においては、複数のチャンネルを有するテレビジョン放送波を受信し、映像信号及び制御信号を出力する複数の受信手段と、前記複数の受信手段により受信された複数の映像信号に係る映像を同一画面上に表示すべく前記複数の映像信号を合成処理して表示部に出力する映像処理手段と、前記受信手段より出力された制御信号に従って前記複数の受信手段と前記映像処理手段とを制御し、前記複数の受信手段により受信すべきチャンネルと前記映像処理手段による表示レイアウトとを自動的に設定する制御手段とを備える構成とした。

【0011】

【発明の実施の形態】 以下、図面を参照して本発明の実施形態について説明する。

【0012】 図1は本発明が適用されるデジタルテレビ受信機100の構成を示した図である。図1において、101は衛星放送受信アンテナ、102、103、10

4はアンテナ101により受信された信号から指定されたチャンネルを選局、検波し、更にデスクランブルを行ってMPEG2のトランスポートストリーム(以下MPEG2-TS)を出力する機能を含むBSデジタルチューナ、105、106、107はデスクランブルされたMPEG2-TSから番組ごとに指定されたビデオストリーム、オーディオストリーム、データ放送データ、および番組配列情報等を抽出するトランスポートデコーダ、108、109はビデオストリームを復号化するビデオデコーダ、110はオーディオストリームを復号化するオーディオデコーダ、111、112は復号化されたビデオデータをフレームごとに格納するビデオメモリ、113復号化されたオーディオ信号に従う音声出力するスピーカ、114はデータ放送画面や画面操作のための文字や図形などの表示データを生成するグラフィック生成部、116はビデオストリームに従う映像データと、グラフィック生成部114により生成された映像データとを一つの表示画面に合成する合成プロセッサ、116は画面表示を行う表示部、117はリモコンと操作ボタンを含む操作入力部、118はユーザからの操作入出力と番組配列情報とに基づいてデジタルテレビ受信機100の機能を制御すると共に、データ放送データ入力によりグラフィック生成部114を制御してデータ放送表示画面を描画する制御部である。

【0013】まず、通常の受信動作について説明する。

【0014】チューナ102によって受信されデスクランブルの結果得られるトランスポートストリームはトランスポートデコーダ105によってビデオストリーム、オーディオストリーム、データストリームが抽出される。ビデオストリームはビデオデコーダ108によってビデオ信号に復号化され、フレームごとにビデオメモリ111に格納され、合成プロセッサ115を通して表示部116に出力される。また、オーディオストリームはオーディオデコーダ110でオーディオ信号に復号化され、スピーカ113から音声として出力される。こうして一つの放送番組が受信され視聴者に提示される。

【0015】また、二つの番組を同時に受信する場合には、チューナ103からトランスポートデコーダ106、ビデオデコーダ109、メモリ112を通して同様に放送番組が受信される。チューナ102で受信されたチャンネルのビデオ信号とチューナ103で受信されたビデオ信号は合成プロセッサ115で合成され、表示部116に同時に表示される。オーディオデータはユーザの操作によってどちらか一方の番組の音声を選択され、スピーカ113から出力される。

【0016】一方、データ放送は、ISO/IEC13818-6に規定されるDSM-CCのデータカルーセル方式により放送局から繰り返し送出される放送番組であり、チューナ104によって受信されて、トランスポートデコーダ107によってデータストリームが抽出され制御部118へ

送られる。なお、本形態では、チューナ104をデータ放送用のチューナとして使用しているが、ビデオデコーダ、オーディオデコーダを設けることにより通常のビデオ、オーディオ受信用のチューナとして用いることも可能である。

【0017】トランスポートデコーダ107からのデータ放送のデータストリームにはテキスト情報、スクリプト情報、画像情報、映像音声情報が含まれ、テキスト情報はW3Cが規定するXML (eXtensible Markup Language) によって記述される。本形態ではデータ放送用XMLの仕様としてHTML4.0をXML1.0で再定式化したXHTML仕様を基本として用いている。XMLでは文書の文字列にタグ(<xxx>と</xxx>に囲まれた部分)によって意味付けされる。また、このタグは「入れ子」構造をとることが可能である。また、表示形式はCSS (Cascading Style Sheet)、スクリプト処理はJavaScriptを用いている。

【0018】これらのデータは制御部118で解釈、実行され、文字、図形、画像情報がグラフィック生成部114で生成され、合成プロセッサ115を通じて表示部116に表示される。音声情報は図示されないサウンド生成部を通じてスピーカ113から出力される。

【0019】図2は制御部118に含まれるCPUで動作するソフトウェアの構造を示した図である。データ放送ブラウザはオペレーティングシステム上で動作するアプリケーションであり、アプリケーションからはグラフィックライブラリとグラフィックドライバを介してグラフィック生成部を制御し文字や図形、画像などを合成プロセッサ115に含まれるグラフィックバッファに対して描画する。またウィンドウ制御ミドルウェアと画面合成部ドライバを介して合成プロセッサ115を制御し、ビデオデコーダ108、109からメモリ111、112を経由して送られる映像と、静止画、文字図形の合成処理を制御する。

【0020】また、トランスポートデコーダドライバを介してトランスポートデコーダが制御され、トランスポートストリームに含まれる番組配列情報を獲得する。そして、番組配列情報から得られる情報に基づき、チューナドライバ、トランスポートデコードドライバを介してチューナ102、103、104およびトランスポートデコーダ105、106、107を制御して目的の放送番組を構成する映像ストリーム、音声ストリームを獲得し、映像デコーダ108、109および音声デコーダ110に送る。

【0021】また、番組配列情報から得られる情報に基づき、チューナ104及びトランスポートデコーダ107を制御し、データ放送データを受信する。さらに操作入出力ドライバを介して操作入出力部117のリモコンおよび操作ボタンとの通信が行われ、ユーザの操作を入力する。

【0022】(データ放送ブラウザの動作) ユーザがテレビ番組視聴中にリモコンや操作ボタンを用いてデータ放送の受信表示を指示すると、制御部118に含まれるCPUはデータ放送受信再生プログラム(ブラウザ)の実行を開始する。起動されたブラウザでは、初めに受信するトランスポートストリームの番組配列情報を取得し、この情報に基づいてトランスポートデコーダ107を制御してデータカルセル方式で送信されてくるデータ放送コンテンツの受信を開始して、画面のXMLデータおよび同データに埋め込まれた画像などの関連データを取得し、制御部118に内蔵するメモリに格納する。

【0023】図3、図4、図5に後述する複数の番組の同時視聴を行うデータ放送のための「受信XMLデータ」の例を示す。図3～図5は1つのXMLデータを示している。このデータ放送は、番組の動作として、他の複数の番組の選局制御とマルチ画面におけるレイアウト制御を含み、マルチ画面テンプレートの機能を果すものである。

【0024】図3、図4、図5のXMLデータには、4つの放送番組の映像を表示する為のオブジェクト301や、音声出力のためのオブジェクト302、画面レイアウトを変更するための操作ガイドを表示するボタンの図形要素等303が含まれている。

【0025】操作ガイドを示す文字や図形要素には表示位置や表示形式を定義するスタイル定義が施されており、例えば303の文字要素には画面の上から680画素、左から360画素のところから、幅500画素、高さ60画素の領域に文字列「ボタンで画面が拡大されます」が配置されるよう属性が与えられている。またXMLデータが起動されたとき初めに実行すべきスクリプトがinitialize()であり、リモコンに設けられた専用のボタン(リモコンには赤、緑、青、黄色の四つの操作ボタンが設けられている)が操作された場合に実行される実行すべきスクリプトが、各々selectRed()、selectGreen()、selectBlue()、selectYellow()であること記述されている。

【0026】ブラウザは図3～図5のXMLデータとこれに関連する画像データなどを取得すると、XMLデータ中の<body>;タグのOnLoad属性に記述されたスクリプト関数を初期スクリプトとして実行する。図3、図4の<script>;</script>;に挟まれた部分にスクリプトが記述しており、この例ではinitialize関数が初期スクリプトとなる。

【0027】本形態の初期スクリプトでは二つのチューナを使って四つの放送番組を受信する制御と、これを画面上にレイアウトする制御を実行する。図6はこの制御の手順を示すフローチャートである。

【0028】番組を受信するためには、先ず受信に使用するチューナを割り当てなければならない(図6のS401)。図3の初期スクリプトではこれをopenTuner()A

PIで実現している。openTuner()APIはブラウザが持つ組み込みAPIで、テレビ受信機で使用可能なチューナに対応したTunerオブジェクトを生成するものである。

【0029】次に番組を受信するために番組局とチャンネルの指定を基に番組を選局する。具体的には、ネットワーク名称“broadcasterA”の指定を、データ放送を含むトランスポートストリーム中のNIT(Network Information Table)情報の中でサーチ(S402)、衛星分配システム情報を獲得する(S403)。ここから得られる周波数などの情報をチューナ102に設定してチャンネル101のデータが多重されたMPEG2-TSを得る(S404)。得られたMPEG2-TSをトランスポートデコーダ105でデコードし、指定されたチャンネル番号“101”のPMTのPIDをPAT(Program Association Table)情報から検出し(S505)、対応するPMT(Program Map Table)情報を獲得する(S406)。

【0030】本形態のブラウザでは、これらの制御はTunerオブジェクトが持つaddService()メソッドで実現される。addService()メソッドは、引数に与えられたネットワーク名称とチャンネル番号で指定されるサービスの受信をTunerオブジェクトに指示し、serviceオブジェクトを生成して戻り値として返す。serviceオブジェクトはサービスを構成する映像、音声、データなどのコンポーネントを持ち、映像の表示や音声の出力にはこれらのコンポーネントを指定する。

【0031】実際のストリームデータの獲得は、PMT情報から得られる映像ストリーム、音声ストリーム、データストリームを指定するPID(Packet Identifier)を用いて、トランスポートデコーダにおけるトランスポートストリームのフィルタリングによって実行される。

【0032】初期スクリプトでは、四つの指定されたチャンネルの受信に加え、これらのチャンネルの映像表示と同時に表示すべきボタン画像(アイコン)の表示位置制御を行っている。即ちsetPositionX(0)関数を呼び、serviceXオブジェクト(Xは表示位置に対応付けられた0～3の数字)とボタンオブジェクトのstyleに表示位置を設定する。表示位置は、serviceが指定する映像オブジェクトのアスペクト比を考慮した位置となっている。また、serviceオブジェクトのsetPosition0()においては可視の属性と、serviceオブジェクトに属するaudioオブジェクトには音声の出力を指定する属性が指定される。図3においては明示されないsetPosition1()からsetPosition3()においては、audioオブジェクトに音声出力を停止する属性が指定される。

【0033】ブラウザは初期スクリプトの実行後、XMLデータの本文部分(図5の<body>;～</body>;部分)を実行する。初めに、オブジェクトとして置かれた四つの映像を表示する。即ち、初期スクリプトで指定された放

送番組を受信して得られるストリームと、その表示位置として指定されたオブジェクトを、合成プロセッサを制御してディスプレイに合成表示する。また、指定されたストリームによる音声を、音声デコードを制御してスピーカに出力し、他の描画要素をグラフィック生成部を制御して描画し、合成プロセッサで合成する。

【0034】図7はこのXMLデータに従う表示画面の例である。図7の501は動画ウィンドウであり、XMLデータで指定された101チャンネルの放送番組の映像が表示されている。また、502、503、504も動画ウィンドウでありXMLデータで指定された161、162、163チャンネルの放送番組の映像が表示されている。506、507、508、509はリモコンに設けられた赤、緑、青、黄色ボタンを表す画像であり、画像210と文字211による説明表示にある通り、リモコンの対応するボタンを押したときに対応する放送番組の映像の表示が左側の大きな画面に移動する事を表現している。

【0035】今、図7の表示状態においてリモコンに設けられた緑色ボタンを押したとき、操作入出力17を通じてブラウザのbodyオブジェクトにonButtonGreenイベントが送られ、XMLデータのselectGreen()スクリプトが実行される。図4に示す通り、selectGreen()スクリプトでは初めにremoval()スクリプトによって画面左に置かれている映像の表示位置を移動し、音声出力を停止する。移動先は、緑色ボタンに対応付けられている161チャンネルの映像(service1オブジェクト)の表示位置である。次にservice1オブジェクトの表示位置を画面左の大きな領域に指定し、音声出力を開始する。

【0036】スクリプト実行後の表示状態を図8に示す。図7において右上に表示されていた161チャンネルの映像が画面左中央に大きく表示され、代わりに101チャンネルの映像が右上位置に移動されている。即ち、リモコンのカラーボタンの操作により、ボタンごとに対応づけられた放送番組の映像を左中央の大きな表示領域に移すことが可能となる。この操作は、テレビ受信機の通常の操作にはないもので、データ放送によるXMLデータを処理することによって実現可能な特有の操作である。以下、この機能をマルチ画面テンプレート機能という。

【0037】BSデジタルテレビ放送のストリームは、放送番組を構成するビデオストリーム、オーディオストリーム、データストリームと、これらのストリームの配列に関する情報である番組配列情報が含まれる。番組配列情報には、自他のネットワーク(放送局)による放送番組予定に関する情報も含まれており、例えば、現在時刻から8日後までの全チャンネルの放送番組に関する放送開始と終了の日時、放送内容を説明する簡単な文字列が含まれる。本形態のデジタルテレビ受信機100においては、リモコンより番組ガイド表示の指示があるとこ

うした番組放送予定のデータを用いて電子番組ガイドを表示することが出来る。

【0038】図9は本形態のテレビ受信機100による電子番組ガイドの表示例である。本形態の電子番組ガイドでは、縦方向に時間経過を、横方向にチャンネルを並び、時間帯ごとの番組の説明を表示している。電子番組ガイドでは現在放送中の番組の選局を行うことが出来る。画面に表示されるカーソル701をリモコンに設けられた方向ボタン等で移動させ、決定ボタンで選択すると、その時点で選択された番組を選局して画面に表示する。

【0039】例えば、現在の時刻を12時過ぎと仮定して、図9の状態でリモコンの決定ボタンを操作すると101チャンネルの番組“Music”が選択され、制御部118は各部を制御して、図10に示す通り、番組“Music”の視聴が行われる。

【0040】電子番組ガイド上で現在放送中ではない将来放送予定の番組を選択した場合には、開始時刻に自動的に選局して表示する「視聴予約」、または適切な録画デバイスを選択して録画の予約を行う「録画予約」の操作が行われる。

【0041】通常の放送番組の場合と同様に、電子番組ガイド上でデータ放送による番組を選択することによりデータ放送番組の視聴を行うことができる。例えば、421チャンネルの番組“観戦ツアー案内”を選択すると、図11のようなデータ放送番組を視聴することが出来る。

【0042】そして、本形態では、図3～図5、あるいは図14、図15に示したXMLデータがこの421チャンネルのデータ放送番組として放送されており、14時過ぎに421チャンネルの番組“極楽ゴルフ三昧”を選択すると、図7で説明したマルチ画面による4番組同時視聴を行うことができる。このとき、この421チャンネルを選択する以外のテレビ受信機の操作、例えば画面表示モードの変更や、複数の表示画面に対する個々の選局など操作は必要ない。

【0043】次に、このようなデータ放送データによるマルチ画面テンプレートの機能を時刻の経過に従って説明する。ここでは、図9の電子番組ガイドが表示する番組の予定に従って番組が編成され放送されるものとする。

【0044】現在時刻が13時過ぎである場合、101、161、162、163、181チャンネルでは通常のテレビ番組が放送され、421チャンネルはマルチ画面テンプレート機能を制御するためのデータ放送番組が放送されている。161、162、163チャンネルの3チャンネルは、一つの放送局で企画された三つの放送番組を放送中であるが、番組名から想像される通り、一つのゴルフ場の三つのホールからの中継を同時に行うものである。

【0045】視聴者はテレビ受信機のチャンネルを161, 162, 163の間で切り替えて、次々に登場する注目選手がプレーするホールを、ゴルフ場で各ホール間を自由に行き来するかのよう視聴することが出来る。このとき、101チャンネルに切り替えれば番組“World News”を、181チャンネルに切り替えれば番組“伊語レッスン”を単独の表示画面で視聴できる。

【0046】また、テレビ受信機100のマルチ画面機能を利用し、同時に視聴したい二つの番組を選局して同時に視聴することも出来る。例えば、テレビ受信機100を2画面表示モードに切り替え、左画面に161チャンネルの画像を表示、右画面に162チャンネルの画像を表示し、必要に応じて画面を選択してチャンネルを切り替える、といった操作を行うこともできる。但し、これらの操作は、テレビ受信機が持つ、マルチ画面テンプレートによらない操作である。そのため、ユーザは2画面表示モードの切り替え、受信すべきチャンネルの指示、表示ウィンドウのレイアウト制御などの操作を行う必要がある。

【0047】一方、421チャンネルのマルチ画面テンプレート機能を持つデータ放送番組“極楽ゴルフ三昧”は、これらの3番組をマルチ画面で表示する一つの放送番組として機能する。

【0048】即ち、図3～図5で説明したXMLデータと同様の動作を行って、161, 162, 163チャンネルで放送される3つの番組の画像を同時に表示する。図12はこのときの表示状態を示すものである。ユーザはチャンネルの変更操作によって421チャンネルの番組“極楽ゴルフ三昧”を選局するだけで、表示画面が3画面となり、各々の画面の選局が自動的に行われる。

【0049】図12において、画面500には3つの映像502, 503, 504が同時に表示され、注目するホールを中継する番組の映像が大きく表示され、この中継番組の音声が出力される。また、リモコンの方向ボタンで注目ホールを移動させ、映像の表示サイズを変更する機能が提供される。即ち、図12においてリモコンの左ボタンを操作することにより、図13に示すように、新しく注目する左側のホールを表示する映像が大きく表示されるようになる。こういった操作はこのテレビ受信機の持つ機能ではなく、データ放送のXMLデータによって放送によって提供される機能である。

【0050】図14、図15は13時～14時の間に421チャンネルで放送されるXMLデータの一部を示すもので、図13と図14は1つのXMLデータを示している。図14、図15では、onButtonLeft、onButtonRightイベントにより画面レイアウト変更のスク립トが実行されていることが判る。

【0051】なお、図9に示した161, 162, 163チャンネルの各ストリームは同一周波数の搬送波を用いて多重されて放送されている。そのため、161, 1

62, 163チャンネルを同時に受信する場合であってもチャンネル間で搬送波を切り替える必要がなく、1つのチューナ、例えばチューナ102を利用して各チャンネルの画像を動画像として表示することができる。

【0052】データ放送番組“極楽ゴルフ三昧”を視聴するうち時刻が14時になったとき、データ放送番組として放送されるXMLデータは図3～図5で示すデータに変更される。即ち、101チャンネルにおいて通常番組“女子プロゴルフツアー中継”の放送が開始され、マルチ画面による同時視聴にこの番組が自動的に追加される。同時に注目番組を拡大表示する操作方法もカラーボタンを用いたものに変更される。

【0053】時刻が15時になると、再び421チャンネルで放送されるXMLデータは変更され、101チャンネルの番組“女子プロゴルフツアー中継”のみが表示され、16時に番組“女子プロゴルフツアー中継”が終了すると共にデータ放送番組も終了する。

【0054】このように時刻と共に変化するXMLデータはデータ放送の番組企画に基づいて生成される。図9の番組“極楽ゴルフ三昧”は、週末の午後、注目されるゴルフ中継の一つもろさず視聴するというコンセプトに基づいて生成される番組であり、異なる時間帯で放送される複数の番組を追いかけようとする選局し、その時点で必要な画面レイアウトを自動的に実現するためのXMLデータをデータ放送番組として放送している。

【0055】そして、テレビ受信機が本来備えていない手段であっても、画面レイアウトごとに最も使いやすい操作を可能とし、その操作方法を画面表示によって視聴者に告知することができる。そのため、視聴者は電子番組ガイドや新聞、雑誌で番組放送予定を精査して目的の番組を見落としなく探し出す苦労から開放され、テレビ受信機のマルチ画面機能を個別に操作する手間もなく、一つのデータ放送番組を選局するだけで、その全機能を自動的に入手することが出来る。

【0056】また、本形態においては表示部116を含むテレビ受信機の構成を示したが、表示部をその構成に含まない、いわゆるセットトップボックスの構成をとることも可能である。その場合、合成プロセッサは接続されたディスプレイで表示するための映像表示信号を出力する機能を含む。その他の構成要素の機能、動作は以上に説明した通りである。

【0057】次に、本発明の第2の実施形態について説明する。

【0058】図16は本発明の第2の実施例におけるテレビ受信機1300の構成を示した図である。図において、1301は地上波テレビ放送受信アンテナ、1302, 1303はアンテナにより受信された信号から選局、検波を行い、ビデオ信号とオーディオ信号を出力する機能を含むアナログテレビチューナ、1304, 1305はNTSC方式によるコンポジットビデオ信号から

輝度信号、色差信号と同期信号を分離し、クロックを生成して、輝度信号、色差信号をA/D変換する機能を含むNTSCプロセッサ、1308、1309はNTSCプロセッサ1304、1305から出力されるデジタルビデオ信号の解像度を表示するサイズに合わせて変換する解像度変換部、1310は入力される二つのデジタルビデオ信号と後述のグラフィック生成部1311で出力されるビデオ信号とを合成して表示部で表示するための映像信号を生成する合成プロセッサ、1312は合成プロセッサで合成された映像を表示する表示部、1306はチューナ1302、1303から出力されるオーディオ信号のバランス、音量、音質などを調整して増幅して出力するオーディオアンプ、1307はオーディオアンプからの出力される増幅された音声信号に基づいて音声出力するスピーカである。また、1313はテレビ受信機の操作ボタンとリモコンを含む操作入力部、1315は図示しない電話回線を経由してネットワークNに接続される、あるいは直接ネットワークNに接続されるネットワークインターフェース、1314は以上に説明したテレビ受信機1300の各部を制御しつつ、操作入力1313から入力されるユーザ操作情報とネットワークインターフェース1315から入力される各種情報に基づいて、テレビ表示に必要な文字図形をグラフィック生成部1311に描画させる機能を有する制御部である。

【0059】図16において、チューナ1302によって受信されたビデオ信号は、NTSCプロセッサ1304でデジタル信号に変換され、適当な解像度変換を経た後、合成プロセッサ1310を通じて表示部1312に出力される。また、チューナ1302によって受信されたオーディオ信号はオーディオアンプ1308で調整、増幅されてスピーカ1307から出力される。この一連の動作により、選局した放送局が放送する一つの放送番組を視聴することが出来る。またテレビ受信機1300は、チューナ1303からNTSCプロセッサ1305、解像度変換部1309を経る第2の受信系を持ち、合成プロセッサ1310における表示映像の切り替えと、オーディオアンプ1306における入力切り替えにより、チューナ1303で受信した別の一つの放送番組を視聴することが出来る。

【0060】また、解像度変換部1308、1309における適当な解像度変換と、合成プロセッサ1310の合成処理により、チューナ1302とチューナ1303で各々受信した二つの放送番組の画像を表示部1312に同時に表示して視聴することができる。このとき、オーディオアンプ1306では視聴者の操作に基づいて、一方の放送番組の音声出力する。図17はこうした2画面表示状態を示すものである。

【0061】図18は制御部1314に含まれるCPUで動作するソフトウェアの構成を示した図である。テレ

ビアプリケーションは操作入出力ドライバを介して得られるユーザ操作情報に基づいて、受信制御ミドルウェアおよび合成制御ミドルウェアに用意されたAPIを利用してテレビ受信機の動作を制御する。APIの呼び出しに基づきオペレーティングシステムの制御下でチューナドライバおよび合成プロセッサドライバを介して、前述の通りテレビ受信機の制御が行われる。またテレビアプリケーションはグラフィックライブラリのAPIを通じて、グラフィックドライバを介して文字、図形、イメージなどをグラフィック生成部に描画させ、合成プロセッサ1310によって合成して画面に表示する。この動作により画面上に受信した放送のチャンネルや、音量調整状態などの情報が表示され、ユーザに通知される。

【0062】テレビアプリケーションは、テレビ受信機の制御部1314に含まれる記憶部に予め格納されたプログラムとして存在し、テレビ受信機1300で実行される組み込みプログラムだけではなく、ネットワークインターフェース1315を通じてネットワークN上の（図示されない）サーバなどからダウンロードされ、制御部1314の記憶部に格納されて実行されるダウンロードプログラムとがある。

【0063】図19、図20はこうしたダウンロードプログラムのソースコードの例を示すものである。図19と図20は一つのソースコードを示している。このプログラムは、時間の経過に従って二つのチューナで独立に二つの番組を同時に受信し、これらの番組の画像を視聴目的に従って表示部1312に合成して表示するマルチ画面テンプレートの機能を持つものである。

【0064】プログラムはJ A V A（登録商標）言語を用いて記述されたGolfParadiseクラスであり、親クラスTvAppletクラスを継承したものである。TvAppletクラスは、ダウンロードプログラムがアプリケーション実行管理下で実行されるために必要な機能を備えたクラスである。即ち、TvAppletクラスにおいてはinit()、start()、stop()、destroy()メソッドを持ち、アプリケーション実行管理がTvAppletクラスオブジェクトをロードした時にinit()メソッドを、実行を開始するときにstart()メソッドを、実行を停止するときにstop()メソッドを、プログラムをアンロードして破棄するときにdestroy()メソッドを呼ぶので、TvAppletクラスの継承クラス（ subclasses ）はその時点で必要な動作をこのメソッドにオーバーライドして記述する。

【0065】また、setAlarm()メソッドを備え、オペレーティングシステムが管理するシステム時刻がsetAlarm()メソッドで指定した時刻と一致したときonAlarm()メソッドを呼び出す機能を有する。TvAppletクラスの継承クラス（ subclasses ）では指定時刻に実行したい制御や動作をこのメソッドにオーバーライドして記述する。更に、操作入力部1313を通じてユーザの操作情報が入力されたとき、onUserOperate()メソッドが呼ばれるの

で、TvAppletクラスの継承クラス(子クラス)はユーザ入力を解析して、必要な動作を行えるようこのメソッドをオーバーライドすればよい。

【0066】また、テレビアプリケーションプログラムはTunerクラスを利用することが出来る。Tunerクラスはテレビ受信機1300の受信動作をオブジェクト化したクラスで、setTune()メソッドにより選局するチャンネルを指定し、setDisplaySize()メソッドにより表示解像度を指定し、setPriority()メソッドにより他の表示オブジェクトとの表示優先度を指定することができる。capture()メソッドは、Tunerオブジェクトがその時点で表示している映像をキャプチャして、その静止画像をImageオブジェクトとして返す。またsetAudioMode()メソッドによりオーディオアンプ1306の制御を行うことが出来る。

【0067】即ち、引数により消音(0)、主音声出力(1)、副音声出力(2)、主+副音声出力(3)が指定される。複数のチューナから出力の指定が同時になされた場合は、両者がミックスされる。また、クラスメソッドopen()メソッドを備え、システムが可能であればTunerクラスのインスタンスを生成して参照を与えてくれる。

【0068】TunerオブジェクトはTvAppletオブジェクトが持つグラフィックオブジェクトTvScreenに描画されことによって画面に映像が表示される。即ちTvScreenクラスのdrawImageメソッドで、描画対象としてのTunerクラスインスタンスと、これを描画する位置とを指定する。TvAppletでは画面表示を行うときpaint()メソッドが呼ばれるので、paint()メソッドで表示したチューナと、他のグラフィック要素を指定して描画する。

【0069】図19、図20においてテレビアプリケーションGolfParadiseクラスの動作を説明するが、ここでは、このプログラムが実行される時間帯に図21のような放送番組が放送されていることを前提として説明する。プログラムはこの放送番組予定を前提として、時間の経過と共にチャンネルを切り替え、複数の番組をマルチ画面として合成して視聴する動作を行う。

【0070】図19において、テレビアプリケーションGolfParadiseクラスのメソッドinit()ではプログラムがロードされたときの初期化を行っている。即ち、チューナのインスタンスtv0を生成し、1チャンネルを選局し、表示解像度を480×360に、表示優先度を0(最優先)に、音声出力を消音状態に指定している。また、もう一つのチューナインスタンスtv1を生成し、4チャンネルを選局し、表示解像度を120×90に、表示優先度を2に、音声出力を消音状態に指定している。最後に状態変数stageを0とし、時刻7時45分にcallbackを指定している。

【0071】メソッドstart()においては、init()メソッドで初期化された状態でプログラムの実行がスタートしたとき、tv1チューナの音声出力されるよう指定し

ている。実際に出力される音量は、このダウンロードプログラムの外側でユーザ操作により決定された音量指定に従う音量である。同時にpaint()メソッドが呼ばれ、状態変数stageの値0に従ってメソッドlayout0()で指定された描画が行われる。

【0072】即ち、後で説明するようにアプリケーション実行管理がダウンロードプログラムを時刻7時30分に起動したとき、チャンネル1の番組「天気予報」とチャンネル4の番組「ノンストップ世界旅行」が図22の如く合成表示され、番組「ノンストップ世界旅行」の音声出力される。

【0073】時刻が7時45分となったときメソッドonAlarm()が呼ばれる。引数timeは現在時刻を示している。if文による分岐に従いtv0の選局がチャンネル2に変更され、時刻による次のcallbackを7時54分に指定する。即ち、図22に示す番組「天気予報」の表示は番組終了とともにチャンネル2で放送中の番組「各地の話題」に切り替わる。

【0074】時刻が7時54分となったときメソッドonAlarm()が呼ばれる。if文の制御に従い、tv1の選局がチャンネル3に変更され、時刻による次のcallbackを7時55分に指定する。同時に状態変数stageの値を1に変更する。即ち、図23に示す通り、番組「世界旅行」の表示に代わって、位置と大きさの違う番組「産業ニュース」の表示が行われる。

【0075】時刻が7時55分となったときメソッドonAlarm()が呼ばれる。if文の制御に従い、tv1の表示がキャプチャされ、その静止画イメージがhighlightに格納される。tv1の選局がチャンネル4に変更され、時刻による次のcallbackを7時55分に指定し、状態変数stageの値を2に変更する。即ち、図24に示す通り、番組「産業ニュース」の中で7時55分に放送された今日のニュースの放送画面がキャプチャされた静止画として表示され、他にチャンネル2の番組「交通情報」とチャンネル4の番組「世界旅行」が受信され、合成されて表示される。

【0076】時刻が8時となったとき、メソッドonAlarm()が呼ばれる。if文の制御に従い、tv0の選局がチャンネル1に変更される。即ち、図24において番組「交通情報」は番組の終了後、チャンネル1の番組「ニュース」に切り替わって表示される。

【0077】本形態においては、実行管理情報ファイルがダウンロードプログラムと一緒にダウンロードされる。実行管理情報ファイルには、ダウンロードプログラムの名称及びファイル名の情報、ダウンロードプログラムの動作を説明する文章を構成するテキスト、ダウンロードプログラムの動作を開始する日付及び時間情報、ダウンロードプログラムの動作を終了する日付及び時間情報、が含まれている。

【0078】図18のアプリケーション実行管理は、ダ

ダウンロードプログラムと実行管理情報ファイルをダウンロードし、ダウンロードプログラムを制御部1314の内部メモリに格納すると共に、実行管理情報ファイルからこれらの情報を集めてテレビアプリケーション実行管理データベースに登録する。組み込みのテレビアプリケーションの一つとして実現される別のアプリケーションにおいて、このデータベースに基づいて、実行可能なテレビアプリケーションを選択肢としてユーザに提示する。

【0079】ユーザの指示により特定のダウンロードアプリケーションの実行が決定されると、アプリケーション実行管理では、実行管理データベースにあるダウンロードアプリケーションの実行開始指定日時に従って、該当するダウンロードアプリケーションの実行を開始する。ダウンロードアプリケーション「GolfParadise」の場合は、前述の通り朝7時30分に実行が開始される。

【0080】以上説明したように、テレビアプリケーション「GolfParadise」はネットワークNを介してテレビ受信機にダウンロードされてテレビ受信機で実行されるダウンロードアプリケーションであり、テレビ受信機1300の複数のチューナ1302、1303による受信機能を制御し、複数の画面を合成表示するマルチ画面におけるレイアウト制御を行い、マルチ画面テンプレート機能を実現するものである。

【0081】本形態で説明したマルチ画面テンプレートは、一例として、視聴者であるユーザに対して出勤前の忙しい時間帯に必要な情報が漏れなく集められるように、複数の放送局から放送される番組を時間の経過と共に順に切り替えて表示するものである。また、同時にリラックスした雰囲気を演出するために、環境ビデオ的な番組を選択してマルチ画面に表示するようテンプレートを構成してもよい。

【0082】こうした番組の組み合わせを決定するためには、新聞や雑誌、その他のメディアを通じて提供される放送番組予定表を視聴の目的に照らして精査する必要があるが、本形態においては、目的に適ったプログラムをネットワークNからダウンロードするだけでテレビ受信機に設定することができる。

【0083】また、高機能なマルチ画面のレイアウトの設定も、テレビ受信機のリモコンや操作ボタンを通じて一つずつ設定する手間も不要となり、高度な機能を最大限に活用することが可能となつて、結果として非常に簡単に、目的に適ったテレビ視聴を行うことができる。

【0084】尚、本形態においては、ダウンロードプログラムの実行開始、実行終了に関する情報をプログラムと同時にダウンロードして、テレビ受信機のアプリケーション実行管理がダウンロードプログラムの実行開始、及び終了を制御するよう説明したが、実行時間管理が可能でTvScreenのような表示オブジェクトによって表示画面を占有しないクラスを持ち、ダウンロード後時刻に関わ

らずこのクラスからプログラムの実行を開始し、視聴開始時刻に実施例で説明したようなTvAppletクラスの子クラスのオブジェクトを起動するような構成をとることも可能である。

【0085】この場合、実行の終了もダウンロードプログラム自身で制御する。いずれの場合も、ユーザによる他の操作によって、実行中のダウンロードプログラムの動作を中止する事が、アプリケーション実行管理の機能として可能である。

【0086】また、本形態においては、ネットワークインターフェースを備えテレビアプリケーションをネットワークNよりダウンロードする構成を説明したが、テレビ受信機がCD-ROMドライブを備え、CD-ROMによってテレビアプリケーションをダウンロードすることも可能である。また、適切なインタフェース手段を備えれば、フロッピディスク、メモリカードなどのメディアをテレビアプリケーションのダウンロード手段に使用することが可能である。

【0087】また、本形態においては表示部1312を含むテレビ受信機の構成を示したが、表示部をその構成に含まず、接続されたディスプレイに映像表示信号出力の機能を含む合成プロセッサの出力によって映像の表示を行う、いわゆるセットトップボックスの構成をとることも可能である。

【0088】更に、本形態におけるチューナはアナログチューナであるとして説明したが、デジタル放送を受信するデジタルチューナであってもよく、デジタル放送によるデータ放送とは別に、ネットワークなどを經由して受信したプログラムによる制御を行うことが可能である。

【0089】また、本発明の各機能をCPUによりソフトウェア処理により実現することも可能である。その場合にも、コンピュータプログラムを記憶したCD-ROMや半導体メモリ、その他の記憶媒体も本発明に含まれる。

【0090】

【発明の効果】以上説明したように、本発明によれば、多くの番組から目的に適う番組を探し出す手間から開放され、また複数の画面ごとに選局と画面レイアウトを指定する操作から開放され、マルチ画面機能を有するテレビ受信機に高度な機能をより簡便に利用することが可能となる。

【図面の簡単な説明】

【図1】本発明が適用されるテレビジョン受信機の構成を示す図である。

【図2】受信動作に関する制御プログラムの様子を示す図である。

【図3】マルチ画面制御に係るXMLデータの様子を示す図である。

【図4】マルチ画面制御に係るXMLデータの様子を示

す図である。

【図5】マルチ画面制御に係るXMLデータの様子を示す図である。

【図6】受信動作を説明するためのフローチャートである。

【図7】マルチ画面テンプレート機能による表示画面の様子を示す図である。

【図8】マルチ画面テンプレート機能による表示画面の様子を示す図である。

【図9】電子番組ガイドの様子を示す図である。

【図10】放送番組の表示画面の様子を示す図である。

【図11】データ放送番組の表示画面の様子を示す図である。

【図12】マルチ画面テンプレート機能による表示画面の様子を示す図である。

【図13】マルチ画面テンプレート機能による表示画面の様子を示す図である。

【図14】マルチ画面制御に係るXMLデータの様子を示す図である。

【図15】マルチ画面制御に係るXMLデータの様子を

示す図である。

【図16】本発明が適用されるテレビジョン受信機の構成を示す図である。

【図17】2画面同時表示モードの際の表示画面の様子を示す図である。

【図18】受信動作に関する制御プログラムを示す図である。

【図19】ダウンロードされる制御プログラムを示す図である。

【図20】ダウンロードされる制御プログラムを示す図である。

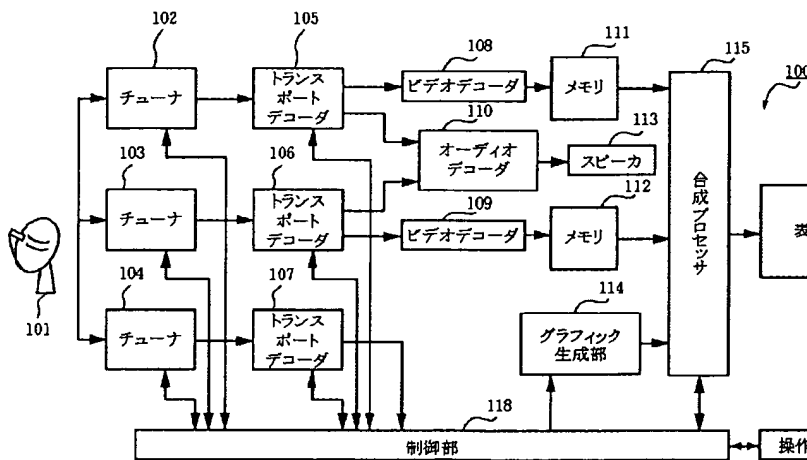
【図21】マルチ画面テンプレート機能によるチャンネル切り替えの様子を示す図である。

【図22】マルチ画面テンプレート機能による表示画面の様子を示す図である。

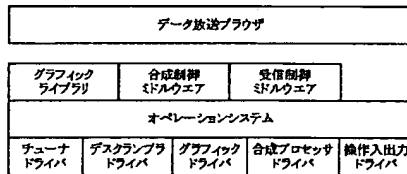
【図23】マルチ画面テンプレート機能による表示画面の様子を示す図である。

【図24】マルチ画面テンプレート機能による表示画面の様子を示す図である。

【図1】



【図2】

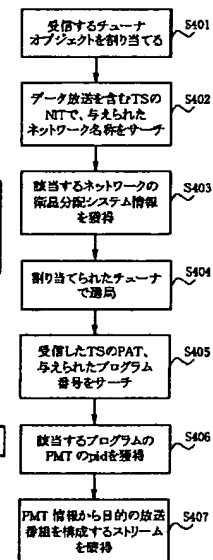


【図9】

電子番組ガイド 701

	101ch	161ch	162ch	163ch	181ch	421ch
12:00-	Music	ヨットレース中継	仏語レッスン		観戦ツアー案内	
13:00-	World News	男子プロゴルフツアー中継	伊語レッスン			
14:00-	女子プロゴルフツアー中継	16H 17H 18H	英語レッスン		極楽ゴルフ三昧	
15:00-		競馬中継	英語レッスン			
16:00-	Movie		日本語教室		ゴルフ教室	
17:00-						

【図6】



【図23】



【 図 3 】

```

<!DOCTYPE dbc SYSTEM "http://www.xxx.co.jp/dbc.dtd">
<head>
<title>Data PARADISE FOR GOLF FREAKS</title>

<script language="JavaScript">

var tuner0, tuner1;           //チューナオブジェクト
var source0, source1, source2, source3; //サービスオブジェクト
var service0, service1, service2, service3; //サービスオブジェクトID
var pos0, pos1, pos2, pos3;    //サービスが置かれている位置
var central;                   //中央位置に置かれているサービス
var btnR, btnG, btnB, btnY;    //ボタンオブジェクト

function initialize(){
    tuner0 = Browser.openTuner();
    source0 = tuner0.addService("broadcasterA", "101");
    service0 = document.getElementById("video0");
    btnR = document.getElementById("buttonRed");

    tuner1 = Browser.openTuner();
    source1 = tuner1.addService("broadcasterB", "161");
    service1 = document.getElementById("video1");
    btnG = document.getElementById("buttonGreen");

    source2 = tuner1.addService("162");
    service2 = document.getElementById("video2");
    btnB = document.getElementById("buttonBlue");

    source3 = tuner1.addService("163");
    service3 = document.getElementById("video3");
    btnY = document.getElementById("buttonYellow");

    setPosition0( 0 );
    setPosition1( 1 );
    setPosition2( 2 );
    setPosition3( 3 );
}

function setPosition0( n ){
    switch( n ){
        case 0:
            service0.style = "position:absolute; top:20; left:40; width:960; height:540; visibility:visible;";
            btnR.style = "position:absolute; top:860; left:70; width:60; height:60; visibility:visible;";
            document.all.audio0.style="sonority:sonorous";
            break;
        case 1:
            service1.style = "position:absolute; top:40; left:40; width:800; height:600; visibility:visible;";
            btnG.style = "position:absolute; top:520; left:70; width:60; height:60; visibility:visible;";
            document.all.audio1.style="sonority:sonorous";
            break;
        case 2:
            service2.style = "position:absolute; top:40; left:40; width:800; height:600; visibility:visible;";
            btnB.style = "position:absolute; top:520; left:70; width:60; height:60; visibility:visible;";
            document.all.audio2.style="sonority:sonorous";
            break;
        case 3:
            service3.style = "position:absolute; top:40; left:40; width:800; height:600; visibility:visible;";
            btnY.style = "position:absolute; top:520; left:70; width:60; height:60; visibility:visible;";
            document.all.audio3.style="sonority:sonorous";
            break;
    }
}

```

【図4】

```

function setPosition1( n ){
    .....
}
function setPosition2( n ){
    .....
}
function setPosition3( n ){
    .....
}

function removal( pos ){
    //あるサービスの現在位置がposのとき (switch文)
    //posがiならば、サービスcentralを位置iへ立ち退かせる
    //サービスcentralの位置に新しい位置iを代入する
    switch( pos ){
        case 0:
            //setPosition0( central );
            //newPos = 0;
            break;
        case 1:
            setPosition1( central );
            newPos = 1;
            break;
        case 2:
            setPosition2( central );
            newPos = 2;
            break;
        case 3:
            setPosition3( central );
            newPos = 3;
            break;
    }
    switch( central ){
        case 0: pos0 = newPos; break;
        case 1: pos1 = newPos; break;
        case 2: pos2 = newPos; break;
        case 3: pos3 = newPos; break;
    }
}

function selectRed(){
    //サービス番号centralとサービス0の位置を交換する
    removal( pos0 );
    setPosition0( 0 );
}

function selectGreen(){
    //サービス番号centralとサービス1の位置を交換する
    removal( pos1 );
    setPosition0( 1 );
}

function selectBlue(){
    //サービス番号centralとサービス2の位置を交換する
    removal( pos2 );
    setPosition0( 2 );
}

function selectYellow(){
    //サービス番号centralとサービス3の位置を交換する
    exchange( pos3 );
    setPosition0( 3 );
}

</script>
</head>

```

【図5】

```

<body onLoad="initialize()"
  onButtonRed="selectRed()"
  onButtonGreen="selectGreen()"
  onButtonBlue="selectBlue()"
  onButtonYellow="selectYellow()"
<div id="div_video0">
  <object id="video0", type="video/mpeg2", source=source0 </object>
</div>
<div id="div_video1">
  <object id="video1", type="video/mpeg2", source=source1 </object>
</div>
<div id="div_video2">
  <object id="video2", type="video/mpeg2", source=source2 </object>
</div>
<div id="div_video3">
  <object id="video3", type="video/mpeg2", source=source3 </object>
</div>

<div id="div_audio0">
  <object id="audio0", type="audio/mpeg2", source=source0 </object>
</div>
<div id="div_audio1">
  <object id="audio1", type="audio/mpeg2", source=source1 </object>
</div>
<div id="div_audio2">
  <object id="audio2", type="audio/mpeg2", source=source2 </object>
</div>
<div id="div_audio3">
  <object id="audio3", type="audio/mpeg2", source=source3 </object>
</div>

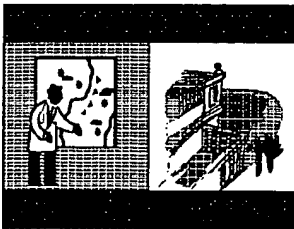
<div style="position:absolute; top:680; left:60; width:300; height:60;">
  <object id="fourButtons", type="image/jpeg" data="fourButtons.jpg"></object>
</div>
<div style="position:absolute; top:680; left:360; width:500; height:60;">
  ボタンで画面が拡大されます
</div>

<div id="div_btnR">
  <object id="buttonRed", type="image/jpeg" data="bntR.jpg"></object>
</div>
<div id="div_btnG">
  <object id="buttonGreen", type="image/jpeg" data="bntG.jpg"></object>
</div>
<div id="div_btnB">
  <object id="buttonBlue", type="image/jpeg" data="bntB.jpg"></object>
</div>
<div id="div_btnY">
  <object id="buttonYellow", type="image/jpeg" data="bntY.jpg"></object>
</div>

</body>

```

【図17】



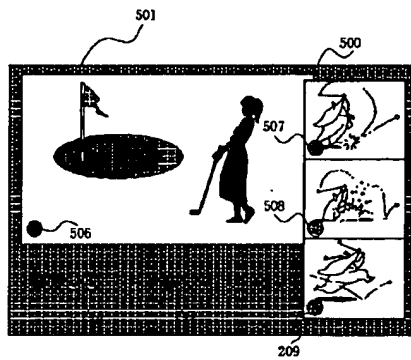
【図18】

テレビアプリケーション				
JVM				
アプリケーション 実行管理	グラフィック ライブラリ	合成制御 ミドルウェア	変換制御 ミドルウェア	
オペレーションシステム				
ネットワーク ドライバ	チューナ ドライバ	グラフィック ドライバ	合成プロセッサ ドライバ	操作入出力 ドライバ

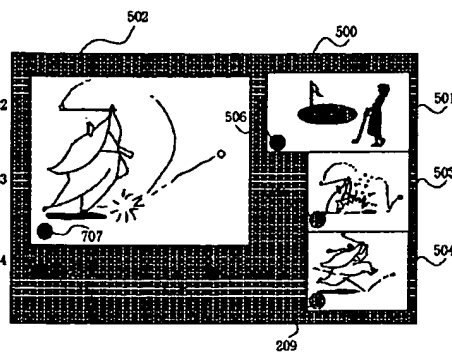
【図21】

	1ch	2ch	3ch	4ch
07:30	天気予報	各地の話題	各地の話題	ノンストップ
07:45	番組案内	交通情報	今日の予定	世界旅行
08:00	ニュース	キャンプへ行こう	マーケット情報	

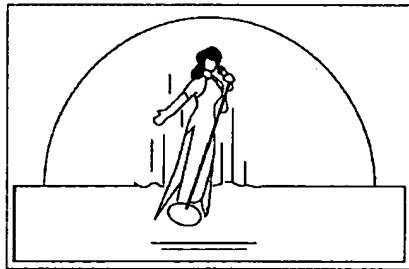
【図7】



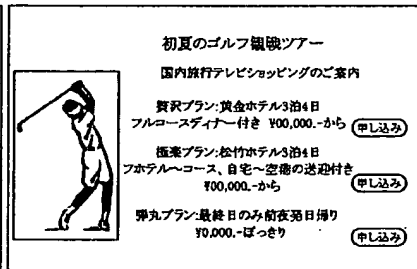
【図8】



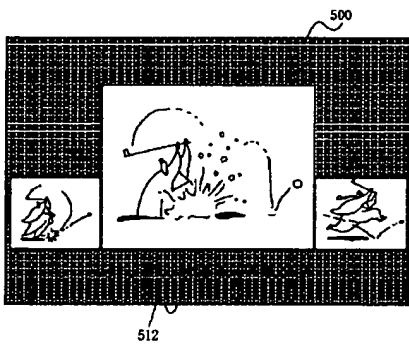
【図10】



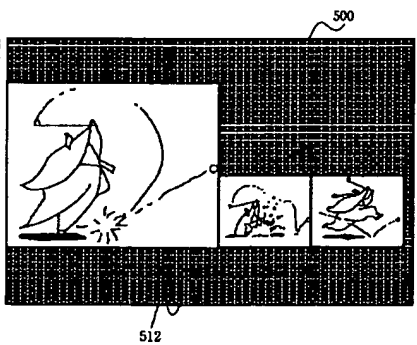
【図11】



【図12】



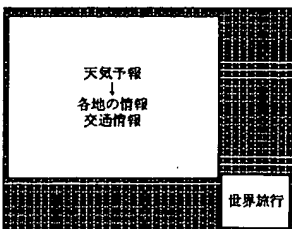
【図13】



【図24】



【図22】



BEST AVAILABLE COPY

【 図 14 】

```
<!DOCTYPE dbc SYSTEM "http://www.xxx.co.jp/dbc.dtd">
<head>
<title>Data PARADISE FOR GOLF FREAKS</title>

<script language="JavaScript">

var tuner0;           //チューナオブジェクト
var source1, source2, source3; //サービスオブジェクト
var service1, service2, service3; //サービスオブジェクトID
var magnified1;        //拡大表示されているサービス

function initialize(){
    ....
}

function setPosition1( n ){
    ....
}
function setPosition2( n ){
    ....
}
function setPosition3( n ){
    ....
}

function selectLeft(){
    //より左の番組を拡大する
    switch( magnified ){
        case 1:
            break;
        case 2:
            setPosition1( 1 );
            setPosition2( 0 );
            magnified = 1;
            break;
        case 3:
            setPosition2( 1 );
            setPosition3( 0 );
            magnified = 2;
            break;
    }
}

function selectRight(){
    //より右の番組を拡大する
    switch( magnified ){
        case 1:
            setPosition1( 0 );
            setPosition2( 1 );
            magnified = 2;
            break;
        case 2:
            setPosition1( 0 );
            setPosition2( 1 );
            magnified = 3;
            break;
        case 3:
            break;
    }
}
```


【図15】

```

removal( pos1 ){
    setPosition0( 1 );
}

</script>
</head>

<body onLoad="initialize()"
    onButtonLeft="selectLeft()"
    onButtonRight="selectRight()">

<div id = "div_video1">
    <object id="video1", type="video/mpeg2", source=source1> </object>
</div>
<div id = "div_video2">
    <object id="video2", type="video/mpeg2", source=source2> </object>
</div>
<div id = "div_video3">
    <object id="video3", type="video/mpeg2", source=source3> </object>
</div>

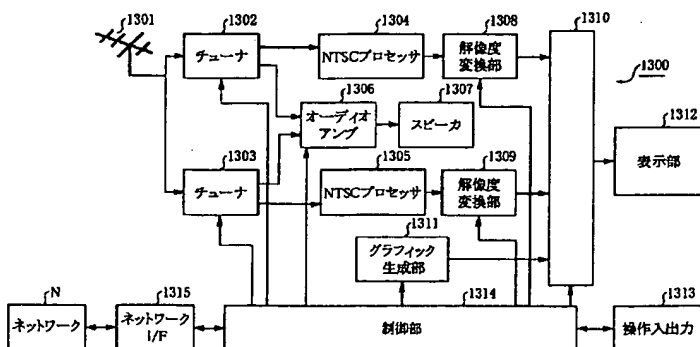
<div id = "div_audio1">
    <object id="audio1", type="audio/mpeg2", source=souroe1> </object>
</div>
<div id = "div_audio2">
    <objcet id="audio2", type="audio/mpeg2", source=source2> </object>
</div>
<div id = "div_audio">
    <object id="audio3", type="audio/mpeg2", souroe=source3> </object>
</div>

<div stile="position:absolute; top:680; left:160; width:700; height:60;"
    ←,→ボタンで拡大画面が切り替わります
</div>

</body>

```

【図16】



【 19 】

```
public class GolParadise extends TvApplet {
```

```
    Tuner tv0, tv1;
    int stage;
    Image highLight;
```

```
    public void init(){
        tv0 = TvAnalogTuner.open();
        add( tv0 );
        tv0.setTune( 1 );
        tv0.setDisplaySize(480,360);
        tv0.setPriority( 1 );
        tv0.setAudioMode( 0 );
```

```
        tv1 = TvAnalogTuner.open();
        add( tv1 );
        tv1.setTune( 4 );
        tv1.setDisplaySize(120,90);
        tv1.setPriority( 2 );
        tv1.setAudioMode( 0 );
```

```
        stage = 0;
        setAlarm("07:45");
```

```
    }
```

```
    public void start(){
        tv1.setAudioMode( 3 );
        validate();
        setVisible( true );
```

```
    }
```

```
    public void stop(){
        setVisible( false );
        tv0.close();
        tv1.close();
```

```
    }
```

```
    public destroy(){
```

```
        .....
```

```
    }
```

```
    private void layout0(TvScreen s){
        s.drawImage(tv1, 510, 380);
        s.drawImage(tv0, 20, 20);
```

```
    }
```

```
    private void layout1(TvScreen s){
        s.drawImage(tv1, 510, 380);
        s.drawImage(tv1, 300, 180);
```

```
    }
```

```
    private void layout2(TvScreen s){
        s.drawImage(tv1, 510, 380);
        s.drawImage(highLight, 300, 180);
        s.drawImage(tv0, 20, 20);
```

```
    }
```

```
    public void Paint(TvScreen s){
        switch( stage ){
            case 0: layout0( s ); break;
            case 1: layout1( s ); break;
            case 2: layout2( s ); break;
```

```
        }
```

```
    }
```

【図20】

```

onAlarm(String time){
    if( time.equals("07:45")){
        tv0.setTune( 2 );
        setAlarm("07:54");
        validate();
    } else if( time.equals("07:54")){
        tv1.setTune( 3 );
        stage= 1;
        setAlarm("07:55");
        validate();
    } else if( time.equals("07:55")){
        highLight = tv1.capture();
        tv1.setTune( 4 );
        stage = 2;
        setAlarm("08:00");
        validate();
    } else if( time.equals("08:00")){
        tv0.setTune( 1 );
        validate();
    }
}

onUserOperate( .... ){
    ....
}

....
}

```

フロントページの続き

(51)Int.Cl. ⁷	識別記号	F I	テーマコード(参考)
H 04 N 5/44		H 04 N 5/44	H
			Z
5/445		5/445	Z
5/60	1 0 1	5/60	1 0 1
7/025		7/08	A
7/03			Z
7/035			
7/08			
7/081			

(72)発明者 吉川 輝樹
 東京都大田区下丸子3丁目30番2号キャノ
 ン株式会社内

(72)発明者 水留 敦
 東京都大田区下丸子3丁目30番2号キャノ
 ン株式会社内

(20) 03-101900 (P2003-[○]混娃

Fターム(参考) 5C023 AA14 AA18 AA38 BA01 BA12
CA03 CA04 CA05 DA04 DA08
5C025 AA23 AA28 AA29 BA18 BA25
BA27 BA28 BA30 CA06 CA10
CA11 CB08 CB09 CB10 DA01
DA04 DA05 DA10
5C026 DA00
5C063 AB03 AB07 AB11 AC01 AC05
AC10 CA11 CA23 CA36 DA03
DA07 DA13 EB32 EB33 EB37
EB38 EB40
5K061 AA09 BB07 BB10 BB19 FF11
GG09 JJ06 JJ07